

时间敏感网络硬件控制点逻辑 HCP 设计方案 (版本 1.0)

OpenTSN

OpenTSN 开源项目组

2021 年 5 月

版本历史

版本	修订时间	修订内容	修订人	文件标识
1.0	2021.5	初版编制		OpenTSN3.0

目录

1、概述	4
2、总体设计	4
2.1. 总体架构	4
2.2 帧的处理流程	7
2.2.1 配置报文解析	7
2.2.2 状态报文上报	8
2.2.3 PTP 封装报文和 ARP 封装报文解封装	9
2.2.2 ARP 请求报文和 PTP 报文报文封装	11
附录 A 数据格式定义	12
附录 B TSMP 消息协议格式	14
附录 C 内部寄存器定义	16
附录 D: command/command_ack 命令格式	22
附录 E: NMAC 报文格式	22

1、概述

OpenTSN2.0 开源硬件逻辑既可作为 TSN 网卡又可作为 TSN 交换机使用，为了简化其作为 TSN 交换机使用时的逻辑复杂度以及增强其作为 TSN 网卡使用时的功能可扩展性，并且考虑到逻辑模块的复用，现将 OpenTSN2.0 开源硬件逻辑拆分为时间敏感网络硬件控制点逻辑（HCP）、时间敏感网络端处理逻辑(TSE)、时间敏感网络交换处理逻辑(TSS)。本文详细介绍时间敏感网络硬件控制点 HCP 逻辑的设计方案，HCP 主要负责配置报文的解析、状态的周期性上报、ARP/PTP 封装与解封装、时钟同步。

2、总体设计

2.1. 总体架构

时间敏感网络硬件控制点逻辑的总体架构框图如图 2-1。

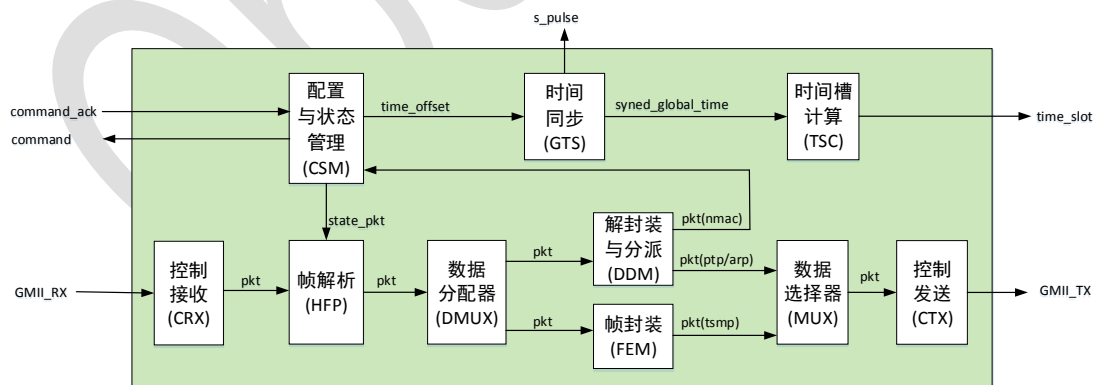


图 2-1 总体架构框图

总体架构图中的信号格式含义如下表 2-1。

表 2-1 HCP 总体架构信号定义

信号	位宽	含义
----	----	----

信号	位宽	含义
pkt（非报文集中缓存模块的信号）	9	报文体数据，具体格式参考附录 A
state_pkt	8	上报状态报文数据
command	204	读、写命令
command_ack	204	读命令响应
time_slot	10	时间槽
time_offset	49	1588 同步从时钟架构需要补偿的值
s_pulse	1	同步脉冲

下面将对整个架构的逻辑模块逐一进行介绍。

CRX（Contorl RX，控制接收）模块：主要功能是接收控制接口传来的报文，将报文传输时钟域从 GMII 接收时钟域切换到架构内部时钟域，以及记录控制接口接收时间同步报文的时间，并存放在 TSNTag 中。根据硬件控制点逻辑所处的阶段来决定是否传输报文：若硬件控制点逻辑处于初始化阶段，丢弃所有报文；若硬件控制点逻辑处于配置阶段，只传输配置报文；若硬件控制点逻辑处于时钟同步阶段，传输非时间敏感报文；若硬件控制点逻辑处于正常工作阶段，接收所有报文。

HFP(Hcp Frame Parse，帧解析)模块：主要功能是从报文中提取以太网类型，将以太网类型和接收时间戳组成报文描述符；对控制接口传来的报文和状态上报报文进行选择，将控制接口传来的报文及其描述符先缓存在队列中，在硬件控制点无上报请求时，将队列中的报文及其描述符读出并输出给下一级模块，若硬件控制点有上报请求且控制接口传来的报文未进行传输时，将状态上报报文输出给下一级模块。

DMUX (DeMultipleXer, 数据分配器) 模块: 主要功能是根据以太网类型对报文进行分派, 并根据硬件控制点所处的阶段决定是否丢弃报文。本模块在初始化阶段将所有报文丢弃, 在配置阶段/时钟同步阶段/正常工作阶段将接收到的 TSMP 报文 (芯片配置帧、PTP 封装帧、ARP 封装帧) 及其描述符分派给解封装与分派模块, 在时钟同步阶段/正常工作阶段将接收到的非 TSMP 报文 (标准以太网非 IP 报文、PTP 报文、NMAC 状态上报报文) 分派给帧封装模块进行封装处理。

DDM (Decapsulation_Dispatch_Module, 解封装与分派) 模块: 将接收到的芯片配置帧、PTP 封装帧、ARP 封装帧分别解封装成 NMAC 配置帧、PTP 帧、ARP 响应帧, 并将 NMAC 配置帧分派至配置与状态管理模块, 记录 PTP 帧发送的时间戳, 将其和 ARP 响应帧分派至数据选择器模块。

FEM (Frame_Encapsulation_Module, 帧封装) 模块: 主要功能是将接收到的 ARP 请求帧、PTP 帧、NMAC 状态上报帧封装到 TSMP 报文中, 针对 PTP 帧, 会记录其接收时间戳、计算透明时钟并累加到透明时钟域。

MUX (MultipleXer, 数据选择器) 模块: 主要功能是对封装后的帧和解封装后的帧进行选择, 并输出到控制发送模块。

CTX (Contorl TX, 控制发送) 模块: 主要功能是计算时间同步报文的透明时钟并更新透明时钟域、将报文传输时钟域从内部处理时钟域切换到 GMII 发送的时钟域、增加帧前导符和帧开始符后将报文

从控制接口输出。

CSM (Configuration and State Manage, 配置与状态管理) 模块: 主要功是对接收到的 NMAC 报文进行解析, 生成写命令; 根据接收到的报文接收/发送/丢弃脉冲进行计数; 以及周期性上报状态。

GTS (Global_Time_Sync, 全局时钟同步) 模块: 主功能是维护一个 48bit 的全局时钟并根据接收到的本地时钟与主时钟的偏移值 offset 对全局时钟进行修正; 维护一个局部时钟, 将局部时钟传给其它模块, 用于记录时间同步报文在硬件控制点的接收/发送时钟, 进而计算时间同步报文在硬件控制点中传输的透明时钟; 维护一个上报周期计数器, 当每经过一个上报周期输出一个上报脉冲信号。

TSC (Time_Slot_Calculation, 时间槽计算) 模块: 根据全局时间、时间槽长度和周期, 计算当前时刻处于一个周期内的哪个时间槽。

2.2 帧的处理流程

硬件控制点逻辑的处理流程包括报文的封装与解封装、配置报文的解析、状态报文的上报。

2.2.1 配置报文解析

HCP 会对芯片配置报文进行解析, 实现对 HCP 和 TSE 或 HCP 和 TSS 中寄存器、表的配置。

控制接收模块接收到芯片配置报文后, 将报文每拍数据的位宽由 8bit 转换为 9bit 后, 对报文进行跨时钟域(从 GMII 接收时钟域到 HCP

内部逻辑工作时钟域) 转换, 并根据 HCP 所处的阶段来对报文进行丢弃或传输: 若 HCP 处于初始化阶段/配置阶段, 丢弃芯片配置报文; 若 HCP 处于时钟同步阶段/正常工作阶段, 传输 TSMP 报文。

帧解析模块从芯片配置报文中提取以太网类型, 将以太网类型和接收时间组成报文描述符, 现将报文及其描述符写到 fifo 中, 在 HCP 无上报请求时, 将 fifo 中的报文和描述符读出并输出给数据分配器。

数据分配器根据报文的以太网类型为 0xff01 判定该报文为 TSMP 报文 (芯片配置报文属于 TSMP 报文中的一种类型), 在 HCP 处于初始化阶段时, 将芯片配置报文丢弃, 在 HCP 处于配置阶段/时钟同步阶段/正常工作阶段, 将芯片配置报文输出给解封装与分派模块。

在解封装与分派模块对芯片配置报文进行解封装与分派。将芯片配置帧解封装为 NMAC 配置帧, 并分派给配置与状态管理模块进行解析。

配置与状态管理模块解析 NMAC 报文, 从 NMAC 报文提取该报文中携带的配置数据个数, 并根据配置的基地址, 将每个配置数据封装都封装到写命令中并输出给端处理逻辑 TSE 或交换处理逻辑 TSS。

2.2.2 状态报文上报

HCP 会对周期性上报端处理逻辑 TSE 和硬件控制点逻辑 HCP, 或交换处理逻辑 TSS 和硬件控制点逻辑 HCP 的状态。

本地时钟每经过一个上报周期, 时钟同步模块会输出一个上报脉冲给配置与状态管理模块, 配置与状态管理模块检测到上报脉冲后,

向帧解析模块发送上报请求，在接收到帧解析模块传来的响应信号后，会根据上报类型，向端处理逻辑 TSE 或交换处理逻辑 TSS 发送读命令，读命令中包含需读取数据的地址；当接收到端处理逻辑 TSE 或交换处理逻辑 TSS 发来的读响应后，从读响应中解析出读数据并将其作为状态报文中的数据输出给帧解析模块。

帧解析模块在监测到配置与状态管理模块传来的上报请求后，当本模块无报文输出时，发送响应给配置与状态管理模块，然后开始将配置与状态管理模块传来的状态报文输出给数据分配器。

数据分配器根据报文的以太网类型为 0x1662 判定该报文为状态上报报文，在 HCP 处于时钟同步阶段/正常工作阶段，将状态上报报文输出给帧封装模块。

帧封装模块将状态上报报文封装到 TSMP 报文中并输出给数据选择器。

数据选择器对解封装后的报文和封装后的报文进行选择，在封装后的报文有传输请求时，将该路报文输出给控制发送模块。

控制发送模块将报文传输时钟域从内部处理时钟域切换到 GMII 发送的时钟域、增加帧前导符和帧开始符后将报文从控制接口输出。

2.2.3 PTP 封装报文和 ARP 封装报文解封装

HCP 会对 PTP 封装报文和 ARP 封装报文进行解封装。

控制接收模块接收到 PTP 封装报文、ARP 封装报文后，将报文每拍数据的位宽由 8bit 转换为 9bit 后，对报文进行跨时钟域(从 GMII

接收时钟域到 HCP 内部逻辑工作时钟域) 转换, 并记录控制接口接收报文的时间, 根据 HCP 所处的阶段来对帧进行丢弃或传输: 若 HCP 处于初始化阶段/配置阶段, 丢弃 PTP 封装报文、ARP 封装报文; 若 HCP 处于时钟同步阶段/正常工作阶段, 传输 PTP 封装报文、ARP 封装报文。

帧解析模块从 PTP 封装报文、ARP 封装报文中提取以太网类型, 将以太网类型和接收时间组成报文描述符, 先将报文及其描述符写到 fifo 中, 在 HCP 无上报请求时, 将 fifo 中的报文和描述符读出并输出给数据分配器。

数据分配器根据报文的以太网类型为 0xff01 判定该报文为 TSMP 报文 (PTP 封装报文、ARP 封装报文属于 TSMP 报文), 在 HCP 处于初始化阶段时, 将 PTP 封装报文、ARP 封装报文丢弃, 在 HCP 处于配置阶段/时钟同步阶段/正常工作阶段, 将 PTP 封装报文、ARP 封装报文输出给解封装与分派模块。

在解封装与分派模块对 PTP 封装报文、ARP 封装报文进行解封装与分派。将 PTP 封装报文、ARP 封装报文分别解封装为 PTP 报文、ARP 响应报文, 并分派给数据选择器; 针对 PTP 封装报文, 会记录 PTP 报文发送时间戳。

数据选择器对解封装后的报文和封装后的报文进行选择, 在解封装后的报文有传输请求时, 将该路报文输出给控制发送模块。

控制发送模块将报文传输时钟域从内部处理时钟域切换到 GMII 发送的时钟域、增加帧前导符和帧开始符后将报文从控制接口输出,

其中针对 PTP 报文,会计算 PTP 报文的透明时钟并更新透明时钟域。

2.2.2 ARP 请求报文和 PTP 报文报文封装

HCP 会对 ARP 请求报文和 PTP 报文进行封装,下面对控制接口接收的 ARP 请求报文、PTP 报文的封装处理流程进行介绍。

控制接收模块接收到 ARP 请求报文和 PTP 报文后,将报文每拍数据的位宽由 8bit 转换为 9bit 后,对报文进行跨时钟域(从 GMII 接收时钟域到 HCP 内部逻辑工作时钟域)转换,并记录控制接口接收报文的时间,根据 HCP 所处的阶段来对帧进行丢弃或传输:若 HCP 处于初始化阶段/配置阶段,丢弃 ARP 请求报文和 PTP 报文;若 HCP 处于时钟同步阶段/正常工作阶段,传输 ARP 请求报文和 PTP 报文。

帧解析模块从 ARP 请求报文和 PTP 报文中提取以太网类型,将以太网类型和接收时间组成报文描述符,先将报文及其描述符写到 fifo 中,在 HCP 无上报请求时,将 fifo 中的报文和描述符读出并输出给数据分配器。

数据分配器根据报文的以太网类型为 0x0806、0x98f7 判定该报文为 ARP 报文、PTP 报文,在 HCP 处于初始化阶段/配置阶段时,将 ARP 报文、PTP 报文丢弃,在 HCP 处于时钟同步阶段/正常工作阶段,将 ARP 报文、PTP 报文输出给帧封装模块。

帧封装模块将状态上报报文封装到 TSMP 报文中并输出给数据选择器,其中针对 PTP 报文,会计算 PTP 报文在 HCP 中传输的透明时钟并更新透明时钟域。

数据选择器对解封装后的报文和封装后的报文进行选择，在封装后的报文有传输请求时，将该路报文输出给控制发送模块。

控制发送模块将报文传输时钟域从内部处理时钟域切换到 GMII 发送的时钟域、增加帧前导符和帧开始符后将报文从控制接口输出。

附录 A 数据格式定义

● 内部传输的 pkt 数据格式

在模块之间传输的报文数据格式为位宽为 9bit。

pkt_data 位宽为 9 位，包含 1bit 头尾标志、8bit 报文数据。头尾标志：1 表示报文头/尾数据；0 标识报文体中间数据。具体如图附 A-1 所示：

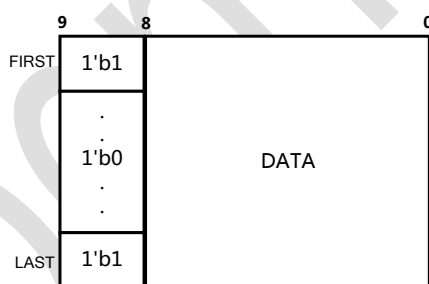


图 A-1 pkt_data 数据格式

● TSNTag 格式

在流量发送端的网卡内部需要根据报文七元组（目的 mac、type、IP 五元组）对时间敏感、带宽预约、尽力转发流量进行分类映射。将分类映射的结果与原报文的 DMAC 字段进行替换，以此进行 TSN 网络的交换，直到接收端的网卡内部进行 DMAC 还原。被替换的 DMAC 字段被定义成 TSNTag。

表 A-1 分类映射关键字 Key

位宽 bit	名称	描述
48	DMAC	报文目的 MAC
16	ETHTYPE	报文以太网类型
8	protocol	报文协议类型
32	Sip	报文源 ip
32	Dip	报文目的 ip
16	Sport	报文源端口
16	Dport	报文目的端口

因同步报文的 TSNTag 中“seq_id”、“frag_id”、“inject_addr”、“submit_addr”信息是无用的，因此可以将时间同步报文的这些字段用来存放架构的接收时间戳信息。而其他非时间同步报文的架构接收时间戳信息是无用的，因此可以延用这些字段的信息。

表附 A-2 时间同步报文的 TSNTag

位宽	名称	位置	描述
3	Flow type	[47:45]	流类型。100：同步报文（其他报文的格式如下表 A-3）
14	Flow id/IMAC	[44:31]	静态流量使用 flowID，每条静态流分配一个唯一 flowID，动态流使用 imac 地址，imac 地址相同的则在交换架构命中同一条表项。
12	Reserve	[30:19]	保留
19	Rx_timestamps	[18:0]	架构接收到时间同步报文的本地时间信息，用于架构发送报文时计算透明时钟。

表附 1-3 非时间同步报文的 TSNTag

位宽	名称	位置	描述
3	Flow type	[47:45]	流类型。 000：ST 分组 001：ST 分组 010：ST 分组 011：RC 分组 101：NMAC 分组 110：BE 分组 111：BE 分组

位宽	名称	位置	描述
14	Flow id/IMAC	[44:31]	静态流量使用 flowID，每条静态流分配一个唯一 flowID，动态流使用 imac 地址，imac 地址相同的则在交换架构命中同一条表项。
16	Seq id	[30:15]	用于标识每条流中报文的序列号
1	Frag flag	[14]	用于标识分片后的尾。0：分片后的中间报文 1：尾拍
4	Frag ID	[13:10]	用于表示当前分片报文在原报文中的分片序列号
5	inject addr	[9:5]	ST 流在源端等待发送调度时缓存地址
5	submit addr	[4:0]	ST 流在终端等待接收调度时缓存地址

附录 B TSMP 消息协议格式

TSMP（时间敏感消息协议）是 TSN 控制器进行网络拓扑探测、对 TSN 芯片和 HCP 进行配置以及对帧进行封装的协议

● TSMP 帧设计原则

- 1) TSMP 帧长度不超过 128B；
- 2) PTP 帧是 TSMP 帧的一种子类型；
- 3) TSNtag 是帧映射后的结果，在 TSN 网络中根据 TSNtag 对帧进行逻辑处理（包括查表转发，入队，调度优先级，ST 流的按时注入、按时提交、输出门控等）；
- 4) 在 TSMP 帧头中设计相关字段用来标识不同类型的 TSMP 帧。

● TSMP 帧格式

TSMP 帧的格式设计如下图所示。

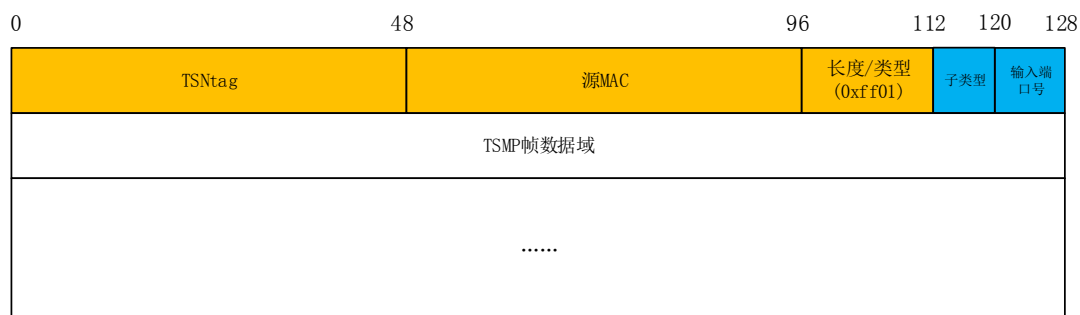


图 B-1 TSMP 帧的格式

图中黄色字段为以太网帧头，蓝色字段为 TSMP 帧头，白色字段为 TSMP 帧数据域。TSMP 帧以太网头和 TSMP 帧头中各字段的含义详见下表。

表附 B-1 TSMP 帧头各字段的含义

字段	位宽	说明
TSNtag	48	TSMP 帧经映 2 射所得的结果。
源 mac	48	暂未使用
长度/类型	16	TSMP 帧类型为 0xff01（自定义）。
子类型	8	用来标识不同类型的 TSMP 帧,目前包含 6 种类型: ARP 封装帧、Beacon 帧、芯片配置帧、HCP 配置帧、ICMP 封装帧、Probe 帧。
输入端口号	8	主机发给 TSN 芯片的帧进入 TSN 芯片的端口号

表附 2-1 TSMP 帧类型

帧类型	子类型的值	含义
ARP 封装帧	8'h0	ARP 帧封装到 TSMP 帧中在网络中进行传输,将 ARP 帧完整地存放在 TSMP 数据域
Beacon 帧	8'h1	交换机、网卡上报到控制器的状态帧,将交换机、网卡的状态上报帧完整地存放在 TSMP 数据域
芯片配置帧	8'h2	控制器对交换机、网卡进行配置的帧,控制器将 NMAC 配置帧封装到 TSMP 帧中,其中 NMAC 配置帧完整地存放在 TSMP 数据域
HCP 配置帧	8'h3	控制器对 HCP 进行配置的帧;配置信息存放在 TSMP 数据域。
HCP 状态上报帧	8'h4	HCP 上报的状态信息存放在 TSMP 数据域

PTP 封装帧	8'h5	将 PTP 帧（sync 帧、delay_req 帧、delay_resp 帧）封装到 TSMP 帧中，其中 PTP 帧完整地存放在 TSMP 数据域
---------	------	---

附录 C 内部寄存器定义

架构内部可配置地址空间主要有两部分，包括：MDID 模块号和真实地址空间，其中 MDID 模块号主要用来区分不同模块，而后 20 位为各个模块使用的地址空间。地址的第 19bit 位用于区别地址类型，控制/表项寄存器可读可写，调试和版本寄存器只读，每个模块的地址空间为 1024k,其中可读可写和只读寄存器各有 512k。具体地址含义如下。

表 C-1 地址格式

ADDR[26:0]		
MDID[26:20]	ADDR[19]	ADDR[18:0]
MDID : 0-127	0	该模块的控制寄存器,表项等,可读可写
	1	只读

每个处理模块的 MDID 号分配如下。

表 C-2 模块中的 MDID 和地址

处理模块	CSM	TIS	TSS	QGC	GTS	FLT
MDID	0x0	0x1	0x2	0x3-0xa	0xb	0xc
地址	0x0-0xffff	0x100000-0x1ffff	0x200000-0x2ffff	0x300000-0xafffff	0xb00000-0xbffff	0xc00000-0xcffff

● CSM 模块

地址范围为 Addr 0x0-0xffff。

表 C-3 CSM 模块寄存器

Addr	Data
------	------

	[31:24]	[23:16]	[15:8]	[7:0]
0x0	offset_l			
0x1	offset_h			
0x2	time_slot			
0x3	cfg_finish			
0x4	port_type			
0x5	qbv_or_ach			
0x6	report_type			
0x7	report_en			
0x8	inject_slot_period			
0x9	submit_slot_period			
0xa	report_period			
0xb	offset_period			
0xc	rc_regulation_value			
0xd	be_regulation_value			
0xe	unmap_regulation_value			
0xf ~ 0xfffff	reserve			

表 C-4 寄存器的具体含义

name	bit	R/W	description	default
offset_l	31:17	R/W	代表时间偏移的高位值的低 15 位，表示毫秒	0
	16:0	R/W	时间偏移的低位，表示拍数	0
offset_h	31:17	R/W	保留位	0
	16	R/W	代表时间偏移的正负值，1 代表正值，如果为 0，则代表负值	0
	15:0	R/W	代表时间偏移的高位值的高 16 位，表示毫秒	0
time_slot	31:11	R/W	保留	
	10:0	R/W	时间槽大小	0
cfg_finish	31::2	R/W	保留	0
	1: 0	R/W	配置完成寄存器，0 代表架构正在初始化，不接收任何报文，	0

name	bit	R/W	description	default
			1 代表初始化完成，可以接收 NMAC 配置报文 2 代表配置完成，可以接收除 ST 报文的任何报文 3 代表可以接收任何报文	
port_type	31:8	R/W	保留	0
	7:0	R/W	网络端口类型寄存器，架构共有 8 个网络端口，寄存器的 0-7 位分别代表 0-7 端口的类型，1 代表非合作类型，处理标准以太网类型的报文，0 代表合作类型，处理 TSN 报文	0
qbv_or_ach	31:1	R/W	保留	0
	0	R/W	调度模式选择信号，网络输出逻辑中的调度机制是 QBV 模式还是 QCH 模式 0 代表 QBV 模式；1 代表 QCH 模式	0
report_type	31:16	R/W	保留	0
	15:0	R/W	上报类型，具体参考附录 D	0
report_en	31:1	R/W	保留	0
	0	R/W	上报使能信号，配置与状态管理模块是否进行周期性上报 0 代表不上报；1 代表上报	0
inject_slot_period	31:12	R/W	保留	0
	10:0	R/W	注入时间槽周期，架构内部时间槽切换的周期值 配置的值范围：1-1024 个	0
submit_slot_period	31:12	R/W	保留	0
	10:0	R/W	提交时间槽周期，架构内部时间槽切换的周期值 配置的值范围：1-1024 个	0
report_period	31:12	R/W	保留	0
	11:0	R/W	上报周期，配置与状态管理模块上报的周期值 配置的值范围：1 (ms) 或 1000 (ms)	0
offset_period	31:24	R/W	保留	0
	23:0	R/W	offset 补偿的配置周期	

name	bit	R/W	description	default
rc_regulation_value	31:9	R/W	保留	0
	8:0	R/W	RC 流的监管阈值，当 BUFID 的剩余个数小于该值，开始丢弃 RC 报文	
be_regulation_value	31:9	R/W	保留	0
	8:0	R/W	BE 流的监管阈值，当 BUFID 的剩余个数小于该值，开始丢弃 BE 报文和 RC 报文	
unmap_regulation_value	31:9	R/W	保留	0
	8:0	R/W	非映射流的监管阈值，当 BUFID 的剩余个数小于该值，开始丢弃非映射报文	
reserve	31:9	R/W	保留	0

● TIS 模块

地址范围为 Addr 0x100000-0x1fffff。

表 C-5 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
send_table_N	ST 报文发送时刻表每项内容，N=0、1、...、1023			
0x100000-0x1003ff	send_table_0 表示第 0 个发送表			
0x100400-0x1fffff	保留			

表 C-6 寄存器的具体含义

name	bit	R/W	description	default
send_table_0	16	R/W	保留	0
	15	R/W	表项有效位，0 代表无效，1 代表有效	0
	14:5	R/W	ST 流在一个应用周期内的注入时间槽	0
	4:0	R/W	TSNTag 中的“send addr”	0
.....				
send_table_1023	16	R/W	保留	0

	15	R/W	表项有效位，0 代表无效，1 代表有效	0
	14:5	R/W	ST 流在一个应用周期内的注入时间槽	0
	4:0	R/W	TSNTag 中的“send addr”	0
0x100400-0x1ffff			保留	

● TSS 模块

地址范围为 Addr 0x200000-0x2ffff。

表 C-7 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
submit_table_N 0x200000-0x2003ff	ST 报文提交时刻表每项内容，N=0、1、...、1023 submit_table_0 表示第 0 个提交表			
0x200400-0x2ffff	保留			

表 C-8 寄存器的具体含义

name	bit	R/W	description	default
submit_table_0	31:16	R/W	保留	0
	15	R/W	表项有效位，0 代表无效，1 代表有效	0
	14:5	R/W	ST 流的提交时间槽	0
	4:0	R/W	TSNTag 中的“send addr”	0
.....				
submit_table_1023	31:16	R/W	保留	0
	15	R/W	表项有效位，0 代表无效，1 代表有效	0
	14:5	R/W	当前 Slot	0
	4:0	R/W	TSNTag 中的“send addr”	0

● QGC 模块

地址范围为 Addr 0x300000-0xafffff，其中 0x300000-0x3ffff 表示第一个端口的门控表，以此类推，共有 8 个端口门控。

表 C-9 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
port0_gate_table_N 0x300000-0x3003ff	0 号端口的门控表, N=0、1、...、1023, 输出门控 port0_gate_table_0 表示 0 号端口的第一个时刻的门控状态			
port1_gate_table_N 0x400000-0x4003ff	1 号端口的门控表, N=0、1、...、1023			
port2_gate_table_N 0x500000-0x5003ff	2 号端口的门控表, N=0、1、...、1023			
port3_gate_table_N 0x600000-0x6003ff	3 号端口的门控表, N=0、1、...、1023			
port4_gate_table_N 0x700000-0x7003ff	4 号端口的门控表, N=0、1、...、1023			
port5_gate_table_N 0x800000-0x8003ff	5 号端口的门控表, N=0、1、...、1023			
port6_gate_table_N 0x900000-0x9003ff	6 号端口的门控表, N=0、1、...、1023			
port7_gate_table_N 0xa00000-0xa003ff	7 号端口的门控表, N=0、1、...、1023			

表 C-10 寄存器的具体含义

name	bit	R/W	description	default
port0_gate_table_0	31:8	R/W	保留	0
	7:0	R/W	0-7 位分别代表 0-7 共 8 个队列的门控状态, 0 代表该队列的门控关闭, 1 代表开启	0
.....				
port7_gate_table_1023	31:8	R/W	保留	0
	7:0	R/W	0-7 位分别代表 0-7 共 8 个队列的门控状态, 0 代表该队列的门控关闭, 1 代表开启	0

● FLT 模块

地址范围为 Addr 0xc00000-0xcfffff。

表 C-11 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
0xc00000-0xc03fff	forward_table_N, 表示转发表, N=0,1,2, ...16383, forward_table_0 表示第 0 个转发表			
0xc04000-0xcfffff	保留			

表 C-12 寄存器的具体含义

name	bit	R/W	description	default
forward_table_0	31:16	R/W	保留	0
	8:0	R/W	转发表的内容，使用 bitmap 的形式，0-8 位分别代表向 0-8 号端口，每位的值 0 代表不向该端口转发，1 代表向该端口转发	0
.....				
forward_table_16383	31:16	R/W	保留	0
	8:0	R/W	转发表的内容，使用 bitmap 的形式，0-8 位分别代表向 0-8 号端口，每位的值为 0 代表不向该端口转发，1 代表向该端口转发	0
0xc04000-0xcfffff			保留	

附录 D: command/command_ack 命令格式

表 D-1 command/command_ack 命令格式

位置	位宽	名称	说明
[203:180]	8	node_id	该字段用来标识对哪个节点进行读写。每个 TSE 或 TSS 都有一个唯一的节点 ID。该字段在 TSN 网卡+TSN 交换机模式下使用到。
[179:172]	8	dest_module_id	该字段用来标识对一个节点内的哪个模块进行控制。TSE 或 TSS 内部每个子模块都有一个唯一的模块 ID
[171:168]	4	type	4'b0001:寄存器或表项的写命令; 4'b0010:寄存器或表项的读命令; 4'b0110:寄存器或表项的读响应。
[167:152]	16	addr	寄存器或表项的读/写地址
[151:0]	152	data	寄存器或表项的读/写数据；其中五元组映射表的表项位宽最大，为 152bit

附录 E: NMAC 报文格式

● 配置报文格式

配置报文格式：在报文中用 count 字段（8bit）表示报文中包含的配置条目数，报文最小为 64 字节，最后不够 64 字节的报文需要补零。NMAC 命令在以太网报文中的封装如图 E-1 所示。

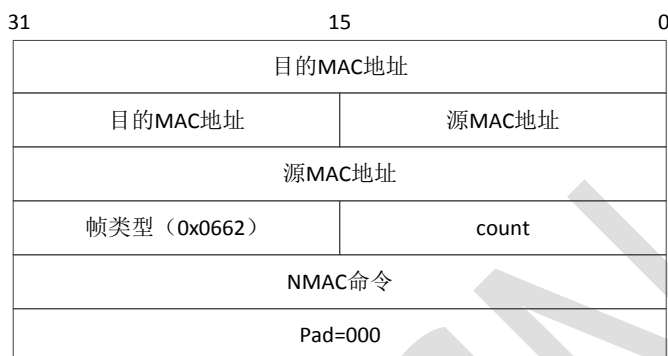


图 E-1 NMAC 配置报文格式

NMAC 命令的格式如图 E-2 所示。

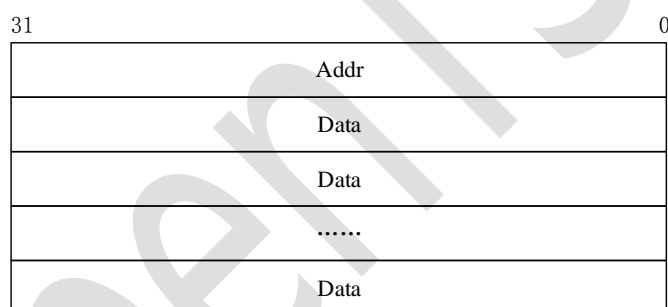


图 E-2 NMAC 命令格式

当配置的寄存器数量为 1，NMAC 命令就包括 32bit 的 ADDR 和 32bit 的 DATA；当配置的寄存器数量为 N（N>1），NMAC 命令就包括 32bit ADDR 和 N*32bit 的 DATA，第一个 DATA 以 ADDR 作为 RAM 写地址，第二个以及后续 DATA 以 ADDR 循环加 1 作为 RAM 写地址。

● 上报报文格式

上报报文格式：其中报文类型为 NMAC 报文（0x1662）。

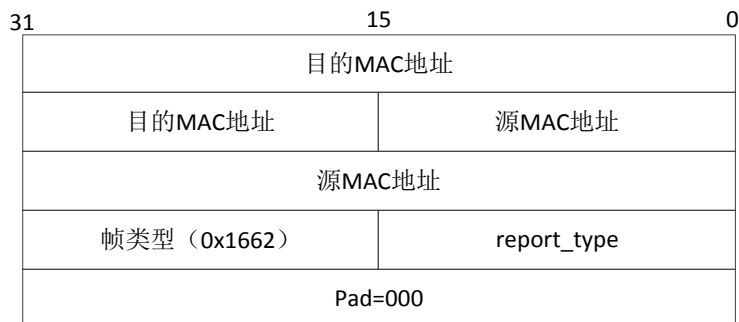


图 E-3 NMAC 上报报文格式

表 E-1 上报类型格式

上报类型 (16bit)		含义
高 6bit	低 10bit	
000000 单个寄存器	0	配置的单个寄存器，包含配置完成寄存器、端口状态寄存器、时间槽大小寄存器、时间偏移寄存器、上报周期寄存器、上报类型寄存器、应用周期寄存器
000001 转发表	0	第 0-63 条转发表
	1	第 64-127 条转发表
	2-255	第 128-16383 条转发表
000010 注入时刻表	0	第 0-63 条注入时刻表
	1	第 64-127 条注入时刻表
	2-15	第 128-1023 条注入时刻表
000011 提交时刻表	0	第 0-63 条注入时刻表
	1	第 64-127 条注入时刻表
	2-15	第 128-1023 条注入时刻表
000100-001011 P0-P7 输出门控表	0	第 0-63 条注入时刻表
	1	第 64-127 条注入时刻表
	2-15	第 128-1023 条注入时刻表
001100 xx_state	0	暂未开发

单个寄存器，report_type 高 6bit 为 0000，低 10bit 为 0。

转发表上报报文格式，report_type 高 6bit 为 000001，低 10bit 为上报的第几块，转发表一共有 16K 条，每条转发表占用 2 字节（低

9bit 有效), 因此每个报文可以携带 64 条, 总共需要 256 个报文。

注入时刻表上报报文格式, report_type 高 6bit 为 000010, 低 10bit 为上报的第几块, 注入时刻表一共有 1024 条, 每条转发表占用 2 字节 (低 9bit 有效), 因此每个报文可以携带 64 条, 总共需要 16 个报文。

提交时刻表上报报文格式, report_type 高 6bit 为 000011, 低 10bit 为上报的第几块, 注入时刻表一共有 1024 条, 每条转发表占用 2 字节 (低 9bit 有效), 因此每个报文可以携带 64 条, 总共需要 16 个报文。

门控表按照端口划分, 每个门控占用一块 RAM, 共有 8 个端口。每个端口两块 RAM, 总共需要 16 块 RAM。